A STUDY OF PARALLEL COMPUTATIONAL MODELS OF VISION

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

H. V. SRINIVASAN

to the

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
MARCH, 1986

CERTIFICATE

This is to certify that the work entitled 'A STUDY OF PARALLEL COMPUTATIONAL MODELS OF VISION' by H.V. Srinivasan has been carried out under my supervision and that this has not been submitted elsewhere for a degree.

(S.K. Mullick)
Professor
Department of Electrical Engineering
Indian Institute of Technology
KANPUR

ACKNOWLEDGEMENTS

I am greatly indebted to my thesis supervisor, Dr. S.K. Mullick for his encouragement and guidance. The amount of freedom and independence that he allowed me in my thesis work was almost unbelievable and enabled me to 'discover' the field of vision for myself and to get a broad background of this subject. Working with him all these days has been a delightful and stimulating experience.

Many others have contributed to some extent or the other at various stages of this thesis.

I would like to thank:

KSA and Venky for their help in the IP Laboratory.
S.N. Pradhan for his excellent typing of this report.
Paddy for helping me in preparing this report.
Poonacha for many useful discussions on various topics, not all of them necessarily connected to this thesis.

Finally, I would like to thank Anand, AV, Bhat, Chitale, Chitnis, Paddy, Phani, Reddy, Rao, TV and many other friends who made my stay here enjoyable and memorable.

H.V. SRINIVASAN

ABSTRACT

This work aims to make a study of some of the parallel computational models of vision. In particular, the pyramids and the parameter nets have been studied in detail. Various image operations were implemented on these computational structures to illustrate some of the important features of these models. The usefulness and importance of such structures is examined in various vision tasks. Biological and Psychological evidences are cited in support of these computational models.

CONTENTS

Page No.

CHAPTER 1

INTRODUCTION

1.1 DEFINING THE FIELD OF COMPUTER VISION:

During the past decade, the field of computer vision has grown into a major subfield of Artificial Intelligence (AI). Now the field spans such diverse disciplines and areas as cognitive psychology, pattern recognition, image processing, computer systems hardware and software, computer graphics, electrical engineering, neurophysiology and mathematics and shares common problems from areas in artificial intelligence including speech recognition, representation of knowledge and robotics. The boundaries of this research area are rather amorphous, particularly if other application domains are included, e.g., bio-medical image processing, industrial automation, military applications and remote sensing.

Broadly stated, vision is the information processing task of understanding a scene from its projected image(s) and the construction of effective computer-based visual systems. However, several fields claim similar tasks as their goal, among them, picture processing, image processing, pattern recognition, scene analysis, image interpretation, optical processing, image understanding etc. These fields overlap to some extent and it is often found convenient to categorize these fields for the purpose of clarifying the goals and

methods of vision research. The categories in which these fields may be grouped are as follows:

1) Signal processing: Signal processing transforms an input image into another image that has desirable properties for example, the output image may have a better signal-to-noise ratio or may be enhanced by emphasising the details to facilitate human inspection. The content of the image is often irrelevant. Image processing and picture processing are the most common terms for this class of processing.

2) Clasification: Classification techniques classify images into predetermined categories. Character recognition is a typical example. Often, a predetermined set of feature values is extracted from images, and the decision of how closely an image 'fits' a class is made on the basis of statistical decision methods applied to multi-dimensional feature space. There is a large body of theory for designing optimal decision rules. These methods are usually called pattern recognition or patter classification schemes.

3) Understanding: Given an image, an image-understanding task aims to build a description not only of the image itself but also of the scene it depicts. In the early years of vision research, the term scene analysis was often used to emphasize the distinction between processing two-dimensional images and three-dimensional scenes. Image understanding requires knowledge about the task world, as well as sophisticated image processing techniques.

1.2 ORGANIZATION OF A GENERAL PURPOSE VISION SYSTEM:

There are many levels of information processing in computer vision. The proper organization of a visual system has been the subject of considerable debate. Issues raised include the controversy over whether processing should be data-driven or goal-driven, serial or parallel, the level at which the transition from iconic to symbolic representation should occur, and whether knowledge should be primarily domain-independent or domain-specifid. With reference to Fig.1, the following are some of the steps in the vision process. Typically, the input to a visual system is by means of some optical sensor and is a two dimensional iconic representation. At the next step features such as edges or segments are extracted from the images, thus obtaining a map-like representation which is variously called as intrinsic image [5] or as a ' primal sketch ' [26] consisting of image features labelled with their property values. The above processes are domain-independent and data driven and are often called low-level processing. Grouping processes may then be used to obtain improved maps from the initial one. The maps may be represented by abstract relational structures by transforming into a representation in which searching and matching against stored models is facilitated. Recognition forms the last stage of visual processing. These are called high-level vision processes and deal with objects and rely on domain-specific knowledge to construct description of scenes.

In other situations , notably in robot vision applications, the scenes to be described are fundamentally three-dimensional, involving substantial object occlusion. Here, the key step in the analysis is to infer the surface orientation at each image point. Two-dimensional segmentation and feature extraction techniques can first be applied to the image to extract such features as surface contours and texture primitives which give an idea of the surface orientation. Using the surface orientation map, feature extraction and segmentation techniques can once again be applied to yield a segmentation into bodies or objects and these in turn can be represented by a relational structure.

Figure 1 shows an ideal two-way process in which knowledge about the expected results of a process can be used to evaluate the actual results and modify the processes so as to improve them. However, in reality the level and extent of interaction of domain-specific knowledge with the various stages of processing is far from clear. The general visual system can be organized into a multilevel, parallel constraint satisfaction model. We will deal with some of these in the subsequent chapters.

1.3 PROBLEMS ENCOUNTERED IN VISION RESEARCH:

We will end this chapter by summarising some of the problems sought to be solved by vision research. Vision is very easy for humans but it is very difficult to construct

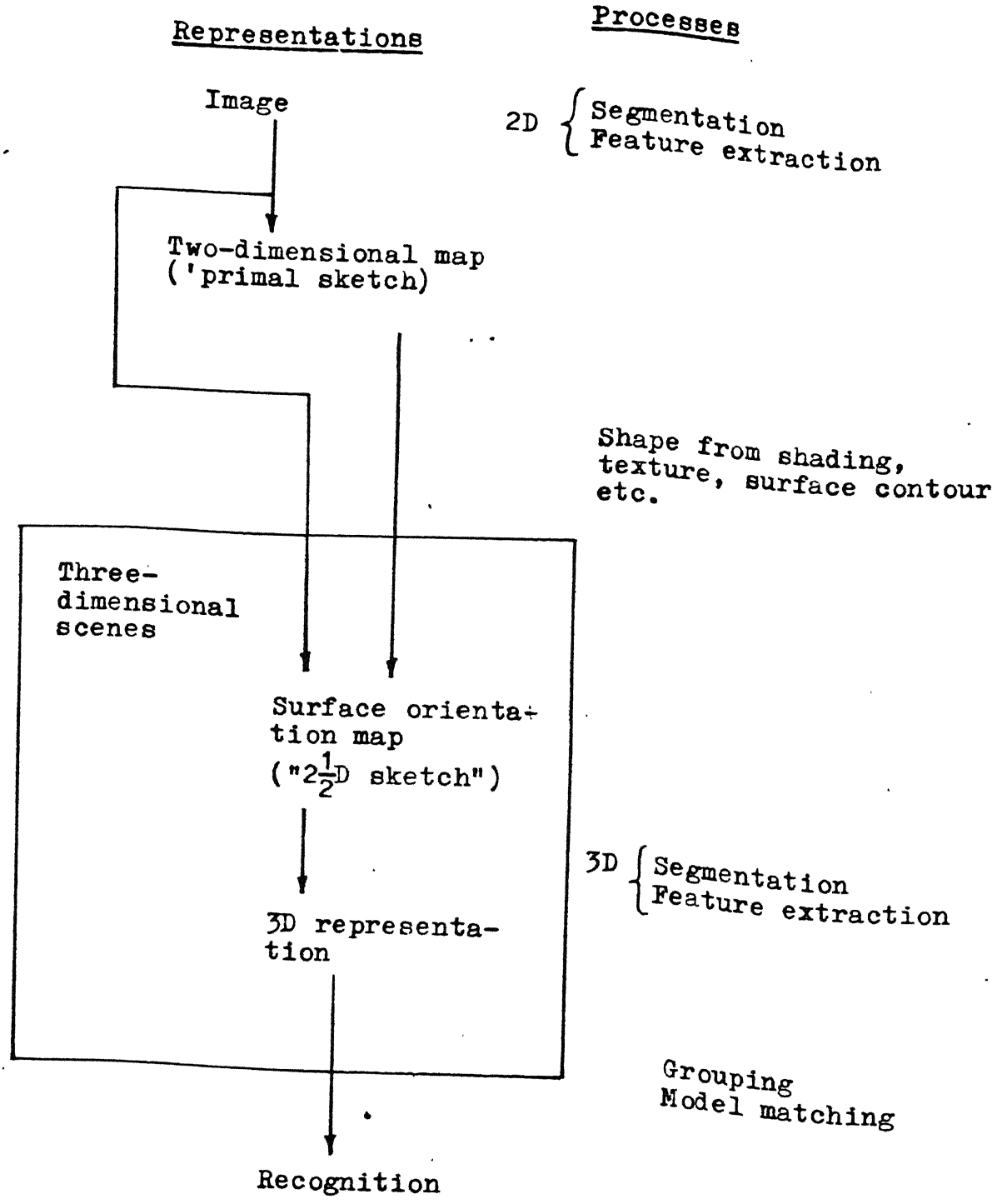


Fig. 1 Simplified diagram of an image analysis system.

a comparable computer-vision system. There are several reasons for this.

First of all, an image underconstrains a scene: It does not provide enough information, by itself, to recover the scene. Among others, the depth information is collapsed by the projection of a three-dimensional scene to a two-dimensional picture. Additional constraints are needed to resolve such ambiguities.

Another reason is that many factors are confounded in an image. The appearance of an object is influenced by its surface material, the atmospheric conditions, the angle of the light source, the ambient light, the camera angle and so on. All of these factors contribute to a single measurement, say, intensity of a pixel. It is difficult to determine the contribution of each factor to a pixel value.

Third, understanding an image requires a priori knowledge of the task domain. For most interpretive tasks, features observable in the image can be weak and cannot be understood without such knowledge. For e.g., an occluded object cannot be recognized unless one knows what one is looking for.

A fourth difficulty with vision research is that the vision task lacks a representation language. Humans can recognize objects with great ease but surprisingly lack a language for description of a scene or shape of an object.

A final problem for vision is the enormous amount of information that is to be computed for even simple tasks. Designing of computer vision system that can execute a task satisfactorily in real-time is still an elusive goal. We will see how this problem can be tackled to some extent by using parallel models in the later chapters.

1.4 OUTLINE OF THE REPORT:

Chapter 2 presents a brief description of some of the important developments in the area of vision. Chapter 3 examines the need for parallel computational models of vision. The basic concepts that are embodied in such structures are highlighted. Chapter 4 deals with the pyramid structure. The way in which segmentation of an image is carried out in a pyramid structure is dealt with in detail. Chapter 5 describes the parameternet formalism. The connectionist theory of perception, the main frame work of the parameter nets, is described. Chapter 6 concludes the report by discussing the merits of parallel computational models of vision and presents in brief the work done . A brief description of shape representation using DOLP transform is given.

CHAPTER 2

VISION:AN OVERVIEW

This chapter briefly describes some of the most important developments in the field of computer vision.

2.1 BLOCKS WORLD UNDERSTANDING:

In the earlier days, a great deal of effort had been expended on the 'blocks' microworld of scenes of polyhedra.

One of the first researchers to be concerned with the recognition of 3-D objects was Roberts [29]. His program understood polyhedral block scenes. A line drawing of the input image was obtained by edge detection and used for recognition. Recognition was carried out by rotating, scaling and projecting a selected model so as to match the input line drawing. Most of the components of today's vision programs-preprocessing, edge detection, construction of line drawings, modelling objects and matching-appeared in this program. However his recognition proceeded sequentially from low to high levels and from image to object. Most of the early work on computer vision took this sequential bottom-up approach.

The first extensive use of heuristics for image understanding was due to Guzman [14]. His program, SEE, could segment a line-drawing into three-dimensional bodies. Guzman classified types of junctions that appear in line drawings and made the important observation that a junction

makes a local suggestion about plausible associations of regions into objects.SEE could correctly segment very complicated line drawing into objects. The serious objections to Guzman's work was that his heuristics had no notion of three dimensional scene features. Besides, his heuristics were very adhoc and had no physical basis.

Much of the difficulties associated with the sequential bottom-up approach to segmentation were sought to be obviated by the induction of knowledge of the task domain. Early work in this line included Falk's INTERPRET [11] which used models to aid interpretation of imperfect line drawings, Yakimovsky and Feldman's semantic based regional analyser [12] which took a decision theoretic approach and Tenenbaum and Barrow's interpretation guided segmentation [30]. These systems for the most part are subject to the same criticism that was levelled against the bottom up systems: They do not distinguish between image characteristics and scene characteristics.

In contrast to the highly heuristic nature of Guzman and Falk, Huffman [19] and Clowes [7] attempted a more systematic approach to polyhedral scene analysis. Most importantly, they emphasised the distinction between the scene domain and the image domain. The scene domain involves physical, three-dimensional aspects of a scene, such as occlusion of one surface by another or the concavity or convexity of edges. The image domain involves the projection

of scene-domain properties on to the two-dimensional picture plane. Huffman and Clowes made the analysis of polyhedral objects more systematic by high-lighting the correspondence between image-domain and scene-domain elements for the polyhedral world.

Waltz [36] extended the research of Huffman and Clowes in two significant ways: first he expanded the set of line labels to include shadows, cracks and separably concave edges. Contrary to expectations, he demonstrated that inclusion of more detailed information does not complicate interpretation but, rather, it constrains and facilitates it. Second, he replaced the simple exhaustive search for consistent line labellings of previous methods by a clever filtering algorithm that examined adjacent junctions in the picture and discarded incompatible candidate labellings.

All the above methods rely on line labelling that characterize the shape only qualitatively. For instance, a line labelling may denote the manner in which two plane surfaces meet but it does not specify anything about the angle at which they meet. Mackworth sought to incorporate the quantitative information into his program POLY [22]. This he achieved by using the concept of what is known as a gradient space. A gradient space comprises of a set of gradients which are a measure of the instantaneous change in the depth of a surface at a point. Using this technique, he could successfully eliminate the problem of some unrealizable legal

labellings that are possible in the other schemes.

Kanade, introduced a representation called the Origami world [21] that admits more objects than the trihedral world does. By using constraints on the gradient space along with junction labelling, Kanade successfully demonstrated the interpretation of many other classes of objects.

2.2 REPRESENTATION OF SCENE CHARACTERISTICS:

In the last ten yearsor so, most vision researchers have abandoned the idea that visual perception can profitably be studied in the context of a priori commitment to a particular program. Efforts have been directed towards extracting scene features from images and have not been restricted to working with the two-dimensional image features.

The most basic features of a scene include orientation, distance from the camera, reflectance and the amount of incident illumination. These are called the intrinsic characteristics. These can be represented iconically, that is, as images. Such images are called intrinsic images [5]. Marr called the intrinsic images by another name- $2\frac{1}{2}$ D sketch [24]. Intrinsic images are most useful because they symbolise important physical features of the scenes.

Perhaps the most fundamental difference between computer vision as it is now and as it was a decade ago, arise from current concentration on topics corresponding to

identifiable modules in the human visual system. Detailed and specific analyses of specific visual abilities are the order of the day. We summarise in brief, some of the research work being pursued in this direction.

Motion: Motion is an important cue that helps in determination of shape of an object and is useful for segmentation of objects. Vision researchers use the term motion to denote multiple images over time. Humans are readily able to perceive three-dimensional structure on the basis of motion information alone. Ullman [35] has tried to duplicate this in a machine with his structure-from-motion theorem. The theorem states that three separate views of four noncoplanar points on a rigid object uniquely define the three-dimensional structure and motion of the object.

Stereo Vision: This technique aims to recover the three-dimensional form of an object by obtaining two views of the object from two different positions and then by measuring the distance from the optical sensor to the object by triangulation. The problem of extracting depth information from a stereo pair of images has, in principle, four components: finding features of an image that are easily recognized in both images, matching these features in the two images, determining the relative camera positions and inferring the distances from the camera to the objects that cast the features in the images. A computational model of human stereo vision has been proposed by Marr et al [25] and uses zero-crossings

in the laplacian image after Gaussian low-pass filtering.

Range Finding: An alternative to inferring three-dimensional structure from two-dimensional data is to measure depth directly Two of the most popular techniques for this are: time-of-flight and triangulation methods. Time-of-flight determines the distance from a source of light (or sound) to an object in terms of the time required for the light or sound to travel to the object and back. Triangulation is another popular technique and as mentioned earlier,is used in stereo vision as well. A detailed survey of the various range-finding methods can be found in [6].

Shape-from Methods: Recently, a class of methods has been developed for recovering shape from shading, textures and contours in monocular images under reasonable assumptions . These methods termed shape-from methods allow us to represent the constraints that images provide and to aggregate them to recover shapes. We briefly touch upon shape-from-shading and shape from texture method.

1) Shape-from shading: Horn [18], under certain assumptions, related the image intensity value (I) at a point in the image to the surface orientation (p,q) at the corresponding three-dimensional scene point. The gradient (p,q) is defined as $p = \frac{\partial f}{\partial x} = \frac{\partial(-z)}{\partial x}$ and $q = \frac{\partial f}{\partial y} = \frac{\partial(-z)}{\partial y}$ where $f(x,y) = -z$ denotes the surface. Thus we have a mapping $f: (p,q) \longrightarrow I$. The task of shape from shading is to find the inverse of this

mapping. In general, the inverse function f^{-1}: I $\longrightarrow$ (p,q) alone does not give a unique orientation for the point. Other constraints, for example, assuming a smooth surface, are needed. Another way of extracting the shape is to obtain an additional image of the same object under different lighting conditions, from which another mapping f_2: (p,q)$\longrightarrow$ I is generated; then both f_1^{-1} and f_2^{-1} may yield a unique solution. This method is called photometric stereo.

2) Shape from Texture:Gradual changes in texture is a cue to depth and information. The two major texture analysis approaches are the statistical and structural ones. The former are generally applicable, the latter can only be used if primitives and,for some methods,placement rules can be extracted also. The various texture analysis methods that are currently popular are well catalogued in [36]. Close parallels can be drawn between shape from shading and shape from texture. We can imagine a small texture element-called a texel-that corresponds to a pixel. The appearance of a texel can be characterized by its shape and local density and varies with surface orientation. Just as shape from shading needed certain assumptions, so does shape from texture. One such assumption is homogenity of surface texture. Additional assumptions of surface uniqueness and continuity are required-as they were for shape from shading-to propagate the constraints and to facilitate the search for a globally correct solution.

In the last decade or so, great significance is attached by researchers in making the links between their work and corresponding theories in psychophysics and neurophysiology more explicit. In large part, this approach stems from the work done by David Marr and his collegues at M.I.T.

Marr's Theory of Vision: Marr's framework of vision is characterized by at least three major processing stages. Each of these stages transforms one representation into another, with the purpose of inferring, and making explicit, relevant information about the surfaces in a scene. The first stage transforms the intensity representations into a primary representation called the primal sketch.[ ] . Changes in the physical properties of surfaces give rise to intensity changes in the images, and it is at the level of the primal sketch that the locations of these changes are made explicit. Obtaining the primal sketch from raw data involves edge-detection and zero-crossings [23]. In the second processing stage, specialized processes, such as those concerned with stereo [25,B3] and motion [35], infer information about the shapes of surfaces from the contents of the primal sketch. Since inferences can be made only at those locations which have been marked in the primal sketch, the information generated is sparse, and it is collected into sparse representations of surface shape that are referred to as

the raw $2\frac{1}{2}$ D sketch. Full surface reconstruction then takes place whereby the sparse representations are transformed into a full $2\frac{1}{2}$ D sketch [B3] containing explicit information about surface shape at all points in the scene, consistent with our perception. The final stage involves computing the 3-D model from a $2\frac{1}{2}$ sketch [25] . The 3-D model computed is an object based representation and facilitates recognition.

The framework for the derivation of shape information from images by Marr's theory can be summarised as shown in Figure 2.

2.3 MARR-HILDRETH'S THEORY OF EDGE DETECTION:

A review of vision techniques will not be complete without a mention of the two most important steps in the early processing of visual data, viz., edge detection and segmentation It will not be possible for us to go into details of various techniques developed for performing edge detection and segmentation. The various edge detection techniques are dealt in detail in [B6]. A survey of the segmentation techniqes can be found in [16]. We will briefly describe the theory of edge detection due to Marr and Hilderth.

The analysis of the edge detection process in treated in two parts.

1) Intensity changes, which occur in a natural image over a wide range of scales, are detected separately at different scales. An appropriate filter for this purpose at a given scale

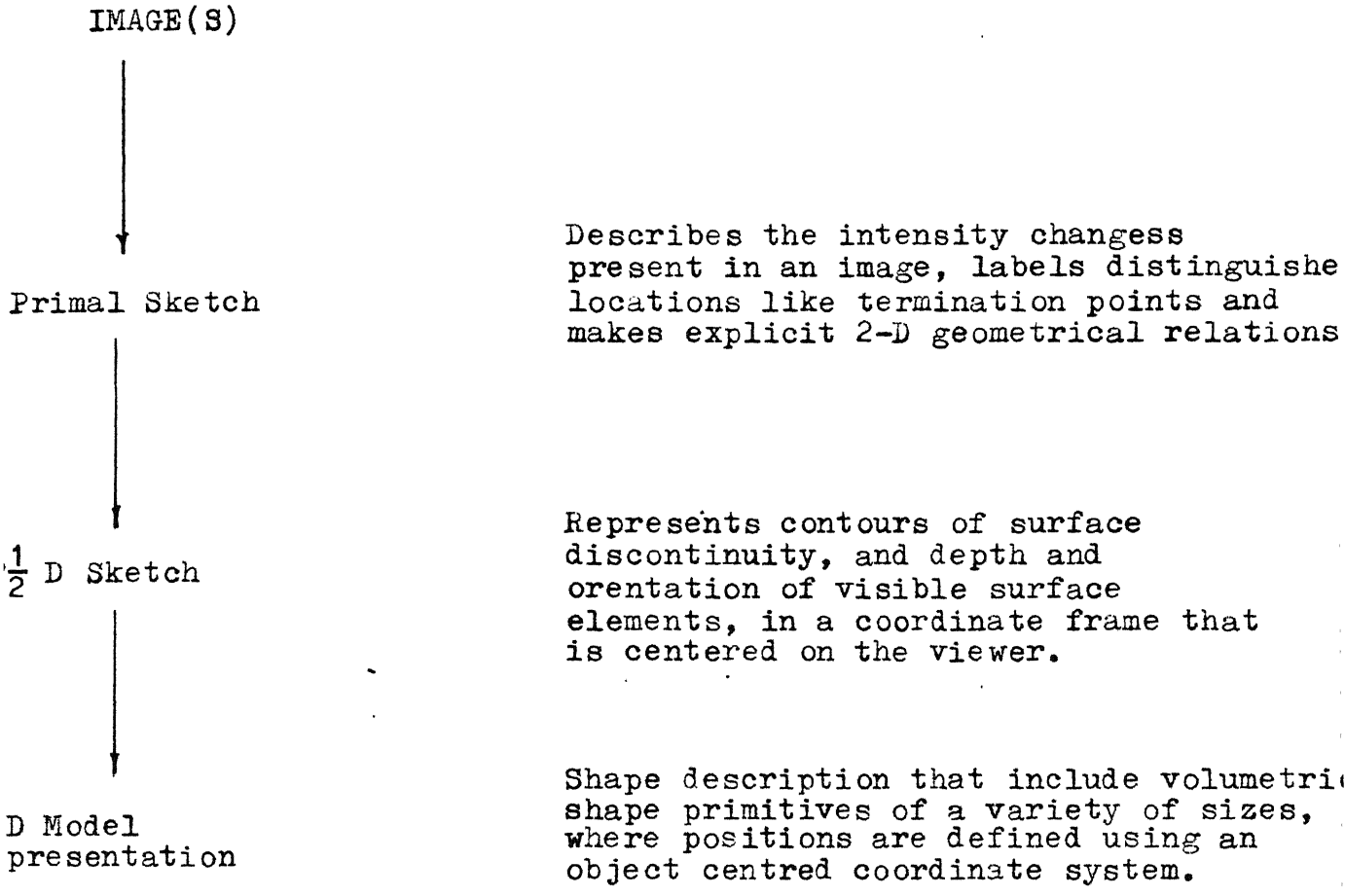


Fig.2 Marr's frame work for the derivation of shape information from images.

is found in the second derivative of a Gaussian, and it has been shown that, provided some simple conditions are satisfied, these primary filters need not be orientation-dependent. Thus, intensity changes at a given scale are detected by finding the zero-values of $\nabla^2 G(x,y) * I(x,y)$ for Image I, where G (x,y) is a two-dimensional Gaussian distribution and ∇^2 is the Laplacian. The intensity changes, thus found, are then represented by oriented primitives called zero-crossing segments. The set of zero-crossing segments together with their amplitudes, constitutes a primitive symbolic representation of the changes taking place within one region of the spectrum of an image.

2) The zero-crossing segments from different channels are combined into a single description called the raw primal sketch. This combining process rests on what is known as the spatial coincidence assumption. Physical edges will produce roughly coincident zero crossings in channels of nearby sizes. The spatial coincidence assumption asserts that the converse of this is true, that is the coincidence of the zero-crossings is sufficient evidence for the existence of a real physical edge. If the zero-crossing in one channel are not consistent with those in the others, they are probably caused by different physical phenomena. The theory also explains several psychophysical findings.

Marr referred to the Logan's theorem as probably indicative of the fact that zero-crossing representation of an image is complete. The Logan's theorem states that under a certain

condition, a one octave band pass signal can be completely recovered from its zero-crossings. The condition imposed is that the signal and its Hilbert transform should have no common zeros [22]. In practice, the $\nabla^2 G$ filter has a half sensitivity bandwidth of about 1.75 octaves. Marr opined that additional information about the slopes of the zero-crossing might help in making the $\nabla^2 G$ filter applicable to Logan's theorem.

It must, however, be pointed out that to date the Logan's theorem has not been extended to two-dimensional signals. Such an extension, even if possible, will be very complicated as pointed out by Logan himself.

2.4 SUMMARY:

An overview of some of the important and significant developments in the area of vision has been made. Only the important features of these developments have been mentioned. Extensive references, quoted all through the chapter, are an excellent source for gleaning additional details.

We have not attempted an over-view of the work-done in three-dimensional object representation Henderson [17] and Besl [6] provide detailed survey of the work in this direction.

CHAPTER 3

PARALLEL MODELS OF VISION : MOTIVATION AND BASIC CONCEPTS

3.1 INTRODUCTION:

The development of computer vision, as we saw in the last chapter, strongly reflects the serial computer for which its programs are coded. The dominant paradigm within AI has been long one of deductive inference, to find a path between some given initial state and some desired goal state, by serially trying different alternatives. This serial approach was so well entrenched that even when models that took on an increasingly parallel_serial approach Such as [30], were proposed, they were not thought of as having this structure.

In the last decade or so, the crucial importance of designing parallel models for vision has been realized.

3.2 NEED FOR PARALLEL MODELS FOR REDUCING PROCESSING TIME:

The enormous amount of data that has to be handled by a vision system makes it imperative for its design to incorporate some amount of parallelism so that processing can be done in a reasonable amount of time, if not in real time. This can be illustrated by the following example:

A television camera inputs a new picture of the scene every 30 milliseconds. Each television picture has about 250,000 primitive spots or pixels in its scan. If a serial computer were to apply an instruction on each pixel

serially , then the total processing time for one picture would be 0.25 Secs (assuming one machine cycle of duration 1 micro second for the instruction). Thus the machine would not have completed its first instruction before the next picture came in .In fact, several different instructions may need to be applied to each pixel for complete processing of the picture.Contrast this with the visual system of a living animal that must always handle things in real time. This means that millions of retinal pieces of information must all be processed within the 10th or 30th of a second before the scane has changed too much, so that smoothly evolving changes can be captured, noted, perceived and acted upon.

3.3 BIOLOGICAL EVIDENCES:

Let us look at the perceptual system in animals to find out whether they are suggestive of a parallel model of processing.

The brain and attendant system of an animal, such as man, that live on the earth is a highly parallel (-serial) system of very roughly 12 billion neurons through which volleys of impulses transfer and process information. The junction where neurons fire into each other, called a 'synapse' can be very complex, with many neurons involved and each neuron has hundreds or thousands of connections with neighbouring neurons (e.g., roughly 38,000 in the visual cortex) [34]. Thus each

neuron is a parallel system in several different ways; it received impulses in parallel from all the neurons that synapse into it; it outputs its impulses in parallel to all the synapses it fires into.

The cerebral cortex, the most ' brainy' part of the brain where perception, perceptual-motor control etc. take place contains roughly 6 billion neurons. It is essentially a thin sheet of about 6 layers of neurons, the sheet crumpled into the familiar form of the brain. Thus some unknown number of transformations of information (no more than six) are made as impulses fire from on side of the cortical sheet to the other. And this goes one in parallel, with a whole region of such columns of neurons firing at the same time. Thus the most plausible picture is one of a structure with many processers interconnected in an enormous network through which impulses flow and fan out, converging and diverging, in a highly parallel (-serial) manner.

The visual system points to another attribute that an ideal visual processing system should probably have. The eye has 120,000,000 rods (the receptors for brightness and motion) and 6,000,000 cones (the receptors of form). Clearly these are all sensing at once, in parallel, so that not the slightestmovement or other change will go unrecorded. But this input transducer layer of sensors is only the first of several highly parallel sets of processors. The retina itself

contains three layers of neurons , forming millions of columns of processors that transform the raw sensed information into gradients and begin to enhance the contours. The optic nerve, which carries information from the eye to the primary visual area of the cortex contains only 1,000, 000 neurons. After the several transformations of information in the retina, the number of elements in parallel has been reduced by a factor of 6 or 10 for the cones and 100 for the rods. From this we can infer one very important aspect of the structure of visual system in living animals, that these systems effect a series of transforms in layers resulting in an overall abstruction and compaction of the information.

Another interesting line of evidence for a high degree of parallel processing can be adduced from a combination of known speeds of neuronal and synaptic transmission and the time it takes for the human beings to perceive. Neurons differ enormously in their dimensions, which determine the amount of time needed for a neuron to transmit an impulse. But in the relatively short neurons of the cortex and the retina the time needed for an impulse to travel through the nerve fiber is relatively short compared to the amount of time for the electro-chemical changes needed to jump the synaptic gap and fire the output neuron [34]. The firing across one synapse typically takes about 1.5 milliseconds, or a bit less. So we can use a figure like 1 to 2 milliseconds to pass through one neuron-synapse step as the brain's basic cycle

time to effect one transform/process. A response to a visual stimulus takes about 100-400 milli seconds depending upon the kind of stimulus and the kind of response. Most of this time is taken by peripheral processes-chemical changes in the cones to sense, conduction of the signal to and from the cortex and innervation and movement of muscles). Thus, there is time for only 10 or 20 or 40 transforms at the most. Actually, a good bit of time must be taken in averaging over inevitably variable and noisy information-that is, integrating and averaging over time-and in combining information and choosing and deciding among alternatives. But the crucial point for this argument is that a small number of transforms can only be assigned as the total set of processes effected by the brain in achieving perception, rather, than the many millions, or billions, of transformation steps that would be needed by a purely serial processor.

3.4 SERIAL VS PARALLEL COMPUTATIONAL MODELS:

Having looked at the need for parallel computation, let us examine the most appropriate structure for a vision system. The strictly serial can take an excessively long amount of time. The strictly parallel can take excessively large number of processors-that is, of material resources and the space needed to contain them. But the strictly serial and the strictly parallel are two extremes. The best bet is to design a system that combines both, i.e. a parallel series system.

To take a simple example, for a completly parallel system to determine whether a field of Os and 1s contain an odd or even number of 1s would need 2^N processors, each with N input connentions to the field (where N is the number of symbols in the field . . In contrast, a completely serial system will need only one processor but at least N units of time. But a parallel-serial system made up of XOR transforms arranged in a pyramid of layers, its decision made by the final processor at the apex of the pyramid, needs only N-1 processor, each with two input connections, and will take only log 2N moments of time. As N grows larger, and time grows more critical, such savings in time can become crucial, with only a small expense in space for processors and connections.

Besides, complete parallel processing of information does not take place in the brain as well. A single binary choice reaction time for a human being (e.g. to a square vs a circle) takes roughly 20 or 40 milliseconds longer than the basic reaction to a single possible stimulus say a light [33]. If the brain were entirely parallel, it would take the same amount of time no matter how complex the problem presented.

3.5 COMMON FEATURES OF PARALLEL SERIES SYSTEMS:

Taking the above facts into consideration, several parallel- series models for vision have been proposed. All these models are remarkable in the fact that they embody the following overall structure.

Refer to Fig. 3. Sensory information is input to the system and stored in an internal memory array. A first set of transformations is effected on this input, and their output is stored in a second internal memory array. A second set of transformations looks at this array, and outputs to a third array. This process continues for as many serial steps as there are layers of transforms. Thus transform-layers are sandwiched between memory-layers. Each layer contains a whole set of transforms, spread about throughout the array. Most transforms look at a relatively local set of information. Usually a single transform will be iterated throughout the array. Often an array will contain several different kinds of transform, e.g., for different sloped and textured edges, each iterated throughout. The transforms in each layer all act at the same time, that is, in parallel. But each layer must wait for and look at the output of the prior layer, thus imposing a serial sequence.Usually, a transform's output memory array will be smaller than its input array, reflecting the degree of abstraction and reduction of information that the transform layer has effected.

3.6 SUMMARY:

The need for parallel processing and the basic features of such parallel models have been discussed in this chapter. The structure of the perceptual system in living animals has been discussed to support the arguments for

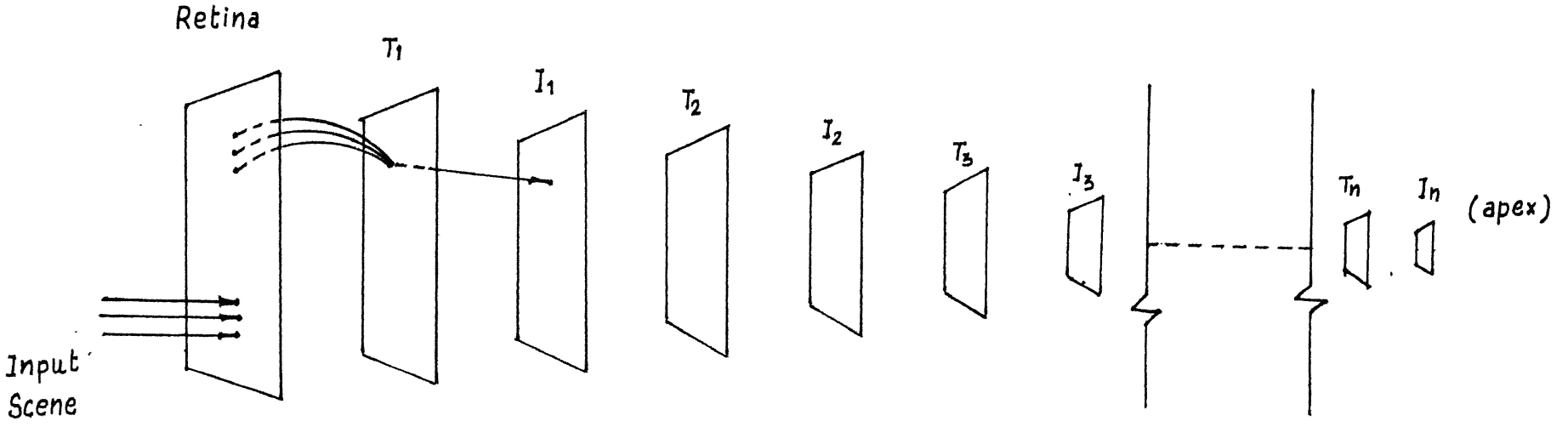


Fig. 3. The overall structure of a parallel series model.

designing parallel (-series) computational models.

The two most important parallel-series hierarchical structures are the pyramids or cones and the parameter nets. We will deal with these in the next two chapters.

CHAPTER 4

PYRAMIDS

4.1 INTRODUCTION :

These structures are variously referred to as Pyramids [7] or as cones [29,35] or, in general, as variable resolution systems. Pyramids are data structures that provide successively condensed representation of the information in an input image. Pyramids may be used to represent information about some desired features in the image.

4.2 USEFUL PROPERTIES OF PYRAMIDS:

Before going into the details of the pyramid structure and the implementation of various image operations on it, it might be useful to summarise some of the useful properties of the Pyramid:

The main advantage of the pyramid is that they provide a possibility for reducing the computational cost of various image-operations using divide-and-conquer principles . For example, intensity-based pyramids can be used efficiently to perform coarse feature-detection operations on an image (at a coarse grid of positions) by applying fine feature detection operators to each level of the pyramid. Pyramids also have the useful property of converting global image features into local features. In particular, they permit

local interactions between feature that are far apart in the original image. Another important way that pyramids can be used is based on establishing local links between nodes at successive levels that represent information derived from approximately the same position in the image.

This provides some interesting possibilities for cooperative computation involving both local and global information, e.g., pixel values and region properties. Pyramids provide a possible bridge between pixel level and region-level image analysis processes.

4.3 THE PYRAMID STRUCTURE:

The pyramid structure can be explained in a formal manner in this way.

An order N pyramid is a layered arrangement of N+1 square arrays A [0], A [1]..., A [N], in which array dimensions are decreased by half from level to level. Refer to Fig. 4. The bottom level measures 2^N by 2^N, the next 2^{N-1} by 2^{N-1} and so on upto the top which has a single node. Thus array A [l] has dimensions $2^{(N-l)}$ by $2^{(N-l)}$, and each array has just 1/4 as many nodes as the next lower array in the pyramid. This decrease in size or spatial resolution from level to level can be effected by requiring adjacent cells at level l to receive data from windows at l-1. These windows may be over lapping [29] or may be adjacent non-over lapping ones [7].

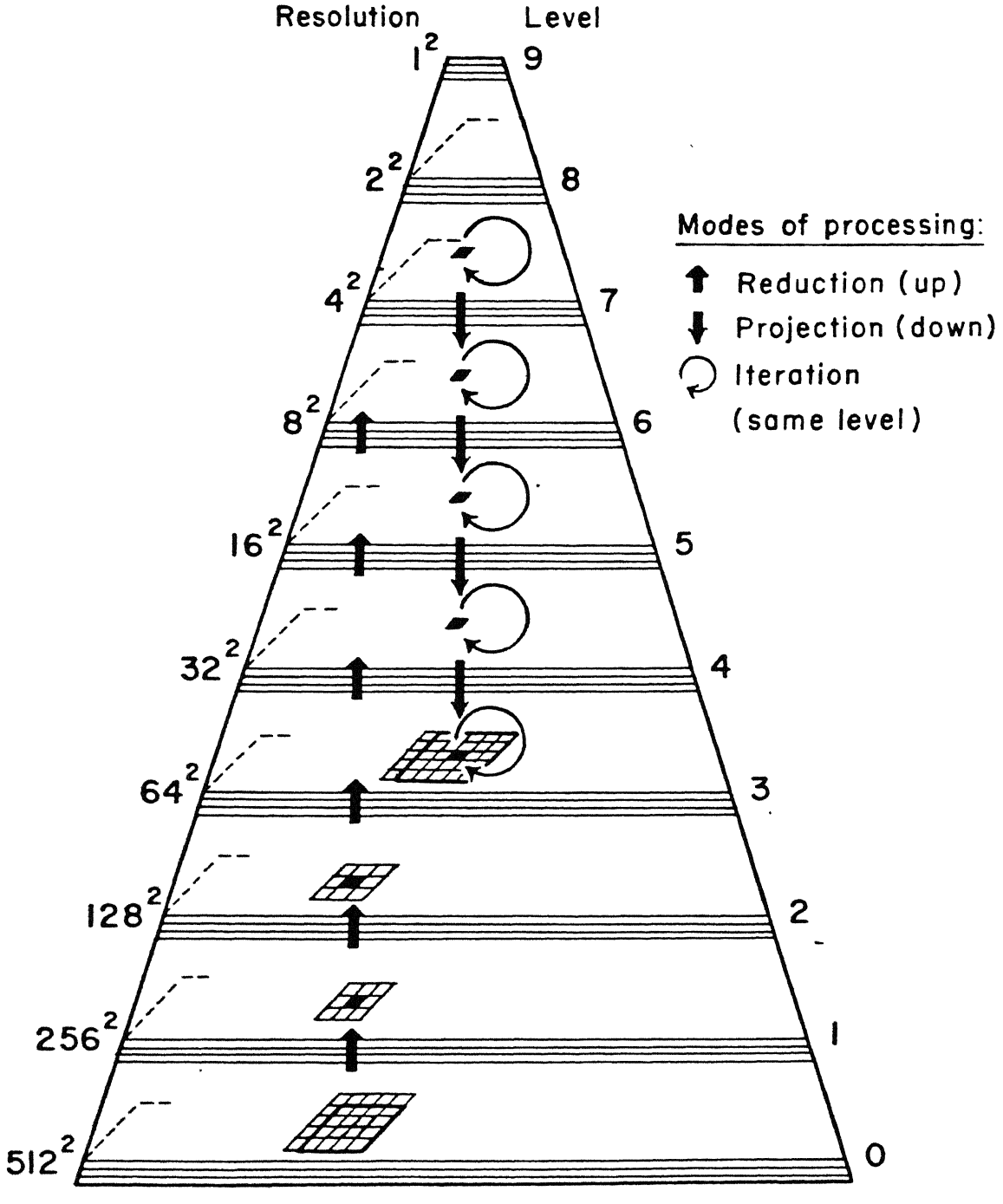


Fig. 4 The Pyramid Structure.

4.4 NON-OVERLAPPING WINDOW TYPE PYRAMID:

Adjacent noeds at level receive data from l-l windows which have non-overlapping, but adjacent 2x2 centers. A father-son relationship may be established between corresponding nodes at adjacent levels. These relations are fixed and do not change from iteration to iteration. This ensures that a window of constant size and shape projects onto a node at the next higher level.

4.5 OVER LAPPING WINDOW-TYPE PYRAMID:

A son-father relationship is defined between nodes in adjacent layers. This relationship is not fixed but may be redefined at each iteration. For each node in level l there is a 4x4 subrray of ' candidate son ' nodes at level l-l. Refer to Fig. 5.

Fig.5 shows arrays at level l and l-l. Only a 2x2 adjacent set of nodes are shown for level l. The corresponding candidate son ' nodes alone are shown in level l-l. Thus the candidate sons of A' is given by the 4x4 block ABCD, that of B' by EFGH, that of C' by IJKLand that of D' by MNOP. It can be seen that two 4x4 blocks of nodes at level l-l correspond to two horizontally adjacent nodes at level l. Note that the blocks overlap 50 percent in the horizontal and vertical directions. It can also be seen that the overlap of all the 4x4 blocks of 'candidate son' nodes result in a block

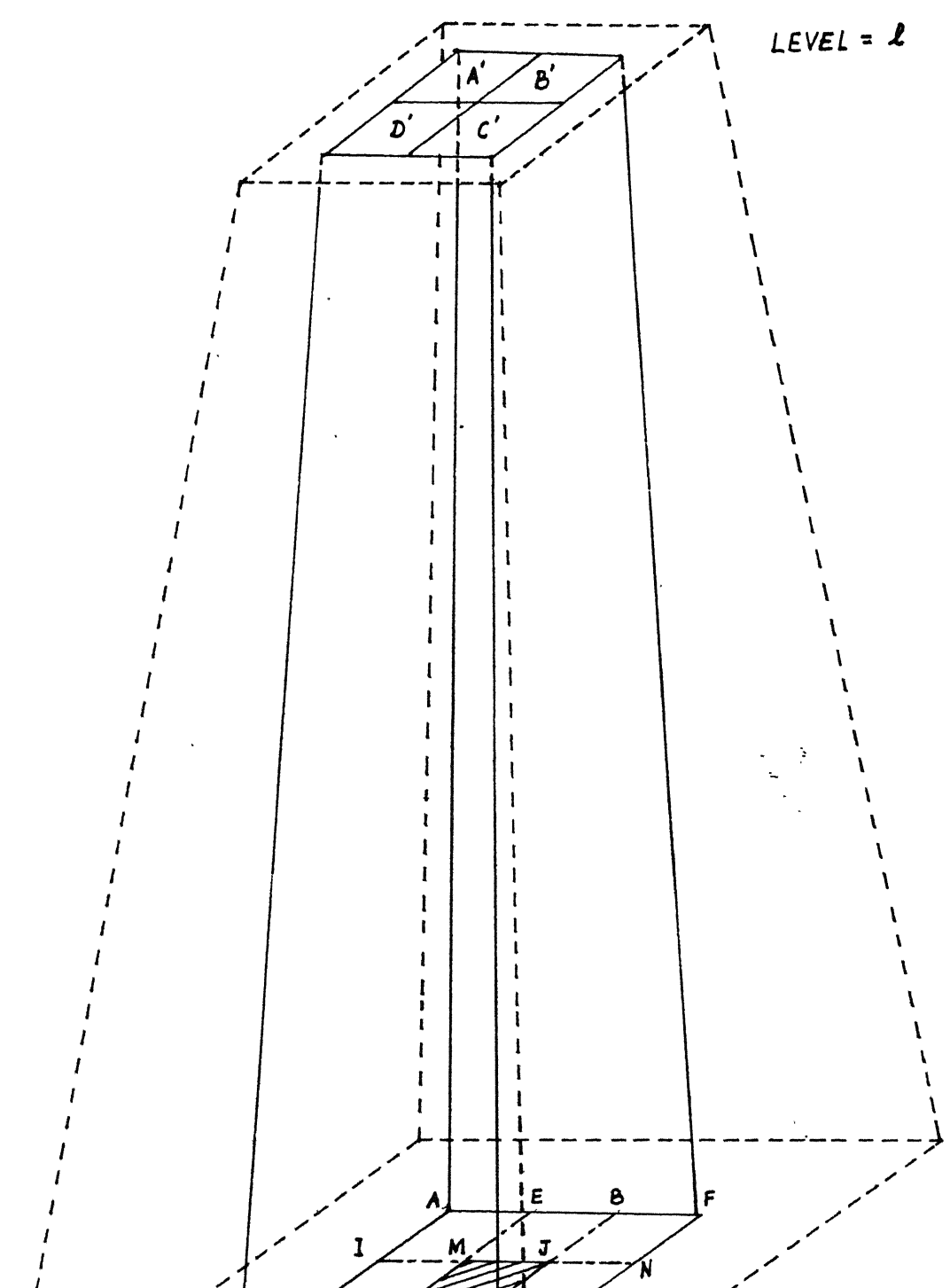


Fig. 5 Overlapping window-type pyramid.

MJCH (the are a shown hatched in the figure). Each node in MJCH is common to all the four nodes A',B',C' and D' at level l. Thus each node is MJCH at level l-1 can be thought of as having four 'candidate fathers' at level l. This can be generalized to all the nodes at level l-1 by giging special consideration to boundary nodes. The special consideration amounts to augmenting each array at level l with an additional row or column on each side. The values assigned to these nodes are reflected from the row or column, one in, from the edge. All nodes at level l-1 may now be thought of as having four ' 'candidate fathers' at level l.

4.6 MODES OF PROCESSING:

There are three basic modes of processing available within the pyramid. They are as follows:

a) Reduction operations in which a window of data at level l is processed and the resultant value(s) stored at l+1.

b) Horizontal operations in which the domain and range of the local function are the same level of the pyramid.

c) Projection operations in which information in the upper layers of the cone influence computations at a lower level.

4.7 SEGMENTATION USING OVERLAP-WINDOW-TYPE PYRAMID:

A variable quantity is associated with each pyramid node. This represents the image property computed within the node's window. The value of the node is typically just the

average of its son's values. Thus the task of computing image properties over windows of many sizes is implemented as a averaging process between pyramid levels.

Image segmentation is implemented as the process which selects a single legitimate father for each node from that node's four candidate fathers. The legitimate father is the candidate with a value most similar to the node itself. If we select any pyramid level L, we find that every bottom level node is linked via intermediate nodes to just one node at level L. Thus these nodes may be regarded as the roots of a number of tree structures within the pyramid which partition the image into segments. Due to the nearest-father selection rule, these segments tend to correspond to homogenous regions.

Let us examine the process of segmentation in greater detail.

We associate four time-dependent variables with each node:

c[i,j,l] [t] the value of the local image property

a[i,j,l] [t] the area over which the property was computed

p[i,j,l] [t] a pointer to the node's father in the next higher array

s[i,j,l] [t] the segment property, the average value for the entire segment containing the node.

Here time t is the iteration number, a positive integer. The c values of the lowest level array are gray samples of the image to be processed.

As noted earlier for each node [i,j,l] above the image level [l>0], there is a 4x4 subarray of candidate son nodes at level l-1. This subarray includes nodes [i,j, l-1], where

$$i' = 2i-1,\ 2i,\ 2i+1 \text{ or } 2i+2$$
$$j' = 2j-1,\ 2j,\ 2j+1 \text{ or } 2j+2 \qquad (1)$$

conversely, each node below the top level [l<N] is a potential son of four level l+1 candidate father nodes [i',j',l+1] where

$$i'' = \{(i-1)/2\} \text{ or } \{(i+1)/2\}$$
$$j'' = \{(j-1)/2\} \text{ or } \{(j+1)/2\} \qquad (2)$$

Here $\{\ \}$ indicates the integer part of the fraction enclosed.

The segmentation process may be viewed of as comprising of four phases. Phase 0 takes place only the initial iteration at t=0 and the computation is bottom up.

Phase 0: The value of c for each level 0 node is set equal to the corresponding image sample value I (x,y) while the c value for each higher level node is an average of all 16 of the node's candidate sons.

For $l = 0$

$$c[i,j,l]\,[0] = I\,(x,y)$$

and for $0 < l \leq L$

$$c[i,j,l]\,[0] = (1/16)\ \Sigma\ c[i',j',l-1]\,[0]$$

All iterations following initialization are divided into three phases:

Phase 1: Son-father links are established for all nodes below the top of the pyramid. Let d (n) be the absolute difference between the c value of node (i,j,l) and its n in candidate father. There are four such candidates. If one d value is less than the other three, then node (i,j,l) is linked to the corresponding 'most similar' father

i.e. if $d\,[m] < d\,[n]$ for all $n \neq m$,

then $p[i,j,l]\,[t] = m$

(The order of numbering the nodes at level l is arbitrary)

If two or more of the distances are equally minimal, then either of the two assignments may be made. If one candidate for which d is minimal is the father of the previous iteration, then this is retained as the updated father. Otherwise one of the equally near candidates is selected at random.

Phase 2:

The c and a values are computed ' bettom up' by reduction operations on the basis of the new son-father links.

For $l = 0$

$a[i,j,0]\ [t] = 1$

$c[i,j,0]\ [t] = I\ (x,y)$

For $0 < l \leq N$

$a[i,j,l]\ [t] = \Sigma\ a[i,j,l-1]\ [t]$

Here, the sum is over this sons of node $[i,j,l]$ as indicated by the links p, assigned in phase 1.

If $a\ [i,j,l]\ [t] > 0$ then

$$c\ [i,j,l]\ [t] = \Sigma\ (a[i',j',l-1]\ [t]\ c\ [i',j',l-1]\ [t])/ a[i,j,l]\ [t]$$

But if $a\ [i,j,l]\ [t] = 0$ so that the node has no sons, $c\ [i,j,l]\ [t]$ is set to the value of one of its candidate sons selected at random.

Phase 3: Segment values are assigned top down. At level L the segment value of each node is set equal to its local property value

$$s[i,j,l]\ [t] = s\ [i'',j'',\ l+1]\ [t]$$

Here node $[\ i'',j'',\ l+1]$ is the father of $[\ i,j,l]$ as established in phase 1.

At the end of phase 3 , the level 0 segment values represent the current state of the smoothing segmentation process. Any changes in pointers in a given iteration will result in changes in the values of local image properties associated with pyramid nodes. These changes may alter the nearest father relationship and necessitate a further adjustment to pointers in the next iteration. Changes always shift the boundaries of segments in a direction which make their contents more homogeneous, so convergence is guaranteed. The iterative processis continued until no changes occur from one iteration to the next.

4.8 OUR IMPLEMENTATION:

For the sake of easiness of description, we had described the pyramid structure as in [30]. Our implementation of the overlapping window-type pyramid differs in certain ways. We define the nodes and their associate properties as two-dimensional variables of i and j only rather than as three-dimensional. The level l is not used as an array index. This ensures saving in space. Besides the arrays at different levels of the pyramid were mapped on to a single output level array of size (N+N/2)x N as shown in Fig. 6.

For a node at level l, the appropriate offsets of the indices are calculated for its sons (at l-1) and fathers (at level l+1) using these offsets, the appropriate property values of each node is mapped on to the output array.

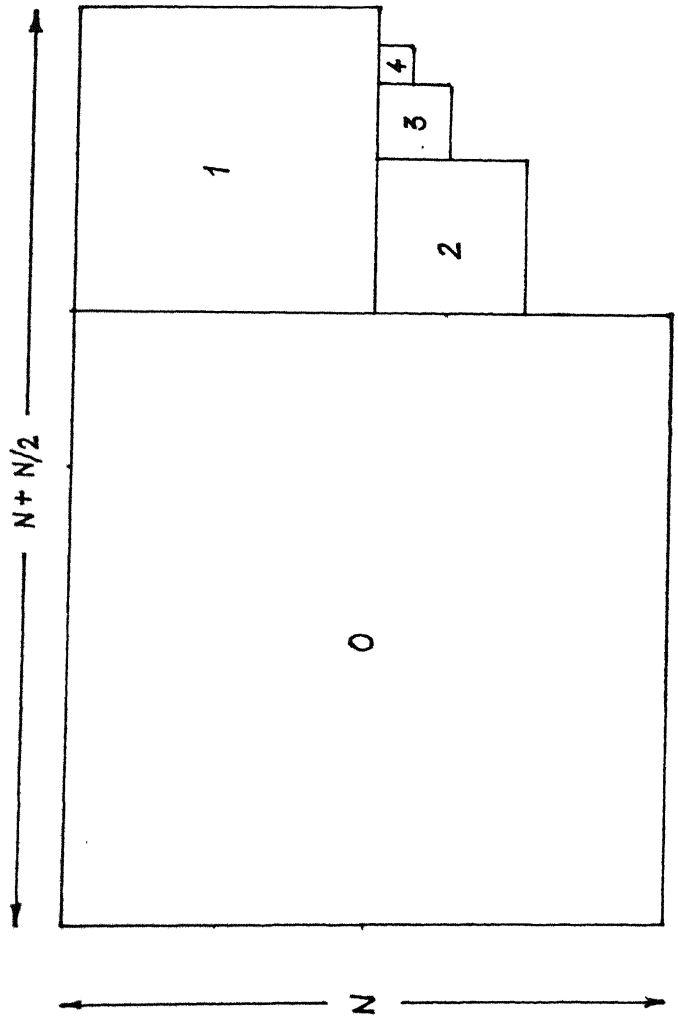


Fig. 6.

4.9 SUMMARY:

The pyramid structure has been described. The characteristics of the pyramid have been high lighted. Segmentation of images on a pyramid has been described in detail.

The pyramid structure can also be used for various other image operations like smoothing, edge detection [29] and texture analysis [28].

CHAPTER 5

PARAMETER NETS

5.1 INTRODUCTION:

Explaining how parts of an image are perceived as a meaningful whole or gestalt is a problem central to vision. A stepping stone towards a solution is recent work on computation of intrinsic images [4,5,19]. Intrinsic images are more amaneable to analysis but they are not grouped into objects. The parameter net provides a model for detection of such groups by representing possible groupings as networks whose nodes signify explicit parameter values.

The nucleus of, what is known as a connectionist theory of low-level and intermediate level vision is sought to be explained by the parameter nets. The theory explains segmentation in terms of massively-parallel cooperative computation between two groups of inter-connected networks. One group, intrinsic images [5], can be computed primarily in terms of local constraints. The other, termed feature spaces, can be computed primarily in terms of global constraints between itself and the intrinsic images. A feature space is also distinguished from intrinsic image space in that it is non-spatio -temporally indexed. Each of these two groups of networks may have many levels of abstraction. Intrinsic images and feature spaces are collectively called

parameter networks because they both have a common organization. The network is an organization of basic units, each representing a value of a particular parameter. The basic element of a parameter network is a parameter unit. A parameter unit will represent a small range of parameter values and has an associated confidence between zero and unity. The values are numerical measurements. Confidence can be (loosely) thought of as a measure of whether or not the value describes the image.

5.2 HOUGH TRANSFORM:

A general way of describing the relationship between parts of an intrinsic image and the associated parameters is a connectionist interpretation of the Hough transform [1,11].

The Hough transform is best illustrated by the case of line detection in edge images. In the parameter net formalism, primal sketch edges such as the line in Fig. 7(a) are represented as parameter units. If there is an edge at (x,y) with orientation α and length s, the vector value of the parameter unit representing the edge is (x,y, α,s). Each such unit is associated with a confidence. For edges, confidence may be initialized to normalized edge magnitude. One way a confidence may be increased is if there are nearby edges of the same orientation which align.

For representation of a feature such as a line, the line may be thought of as composed of a number of edge

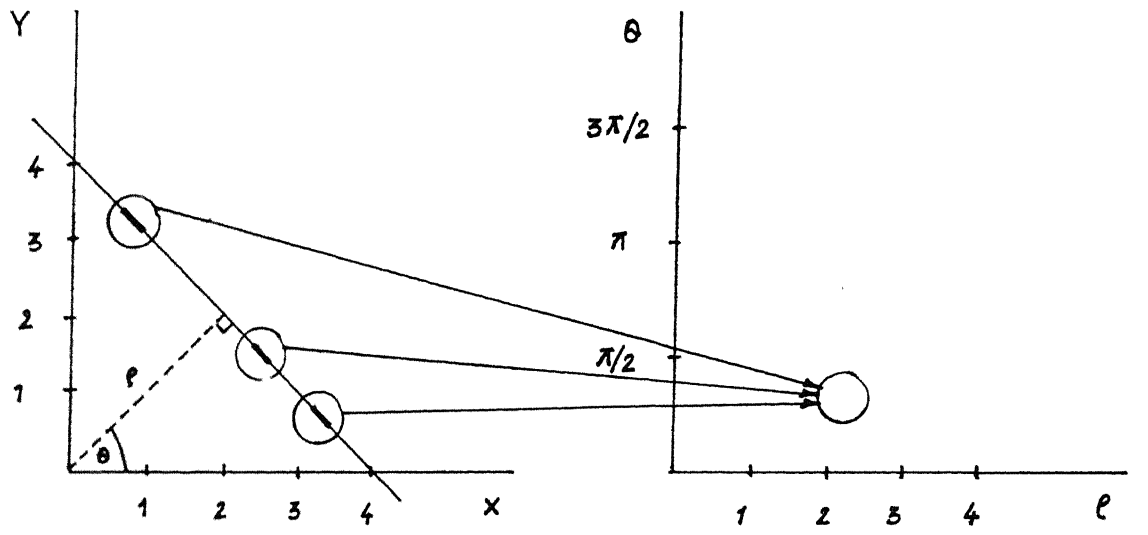


Fig. 7

units. With respect to the edge unit, intrinsic image lines are a global feature. The edge units of a line can be mapped on to a (p,θ) feature space where p represents lines as a distance from the origin p and a orientation θ where, for a point (x,y) on the line, $x \cos \theta + y \sin \theta = p$. Each high confidence edge unit raises the confidence of a corresponding line unit (p,θ) by virtue of incrementing an accumulator. The Hough algorithm for line detection can be summarised as follows:

1) Quantize parameter space between appropriate maximum and minimum values for p and θ.

2) Form an accumulator array $A(p,\theta)$ whose elements are initially zero.

3) For each point (x,y) in a gradiant image such that the strength of the gradient exceeds some threshold, increment all points in the accumulator array along the appropriate line, i.e., $A(p,\theta)=A(p,\theta)+1$.

4) Local maxima in the accumulator array now correspond to collinear points in the image array.

A nice metaphor for understanding the Hough transform is that of voting. Intrinsic image data votes for consistent values of global parameters. The parameter value with the most votes is selected as the global feature space. Selecting the maximum vote feature is equivalent to computing the mode of a distribution in feature space and this makes it easy to understand why the Hough transform is so resilient

to noise and occlusion: only a relatively significant number of model points need be present.

The Hough transform has been generalized to detect arbitrary shapes [1]. As we will see later, this makes the Hough transform especially useful for object recognition.

5.3 SEGMENTATION AS SIMILAR PARAMETER VALUES IN FEATURE SPACE:

In the parameter net architecture, the Hough transform has additional importance as it defines connections between units. These connection patterns are termed constraint maps. The example of line detection uses a constraint map between edge units.and line units. A patch of red in an image may be seen as a unit by a constraint map. For this to happen, an association is made between red points in the image and the particular value 'red' in a parameter space of colors. The constraint map has three elements, a less abstract network of units $\underline{a}$, where

$$\underline{a} = (x, y, r(x,y), b(x,y), g(x,y))$$

a more abstract network of units given by

$$\underline{b} = (r, g, b)$$

and the relationship between the two networks of units, $f = \underline{a}' - \underline{b}$ where a' = the last three components of a .

In other words, each spatial color unit is connected to its non-spatial counterpart. The confidence of a red

non-spatial unit will be high if there are several high-confidence spatially indexed red units.

In the parameter net formalism, parts of an image are perceived as a segment if each of the image parts have the same set of parameter values in a feature space. The general idea is illustrated by the following examples:

1) Parts of a color image may be seen as a segment if they have the same hue. In this case, the feature space is a space of colors and the parts connect to a common unit representing the common hue.

2) Parts of an optical flow image may be seen as a segment if they are part of a rigid body that is moving. In this case, the feature space represents the rigid body motion parameters of translational and rotational velocity and parts of the image connect to a common unit in that space.

3) Parts of edge and surface orientation images may be seen as a segment if they are part of the same shape.

5.4 SHAPE RECOGNITION USING HOUGH TRANSFORM:

We will discuss recognition of shapes by using Hough transform briefly. The general 3-D object recognition task can be thought of as building a description of the object that must have at least two parts: 1) the internal description of the object itself (with respect to an object-contered frame); and (2) the transformation of the object-centered frame to the viewer-centered (image) frame. The reason for this decomposition

is parsimony, different views of the object should have minimal impact on its description. An instance of an object in the viewer centered frame may be related to its internal representation by a viewing transformation. This transformation is completely specified in the general case by seven parameters - three for translation, three for orientation and one for scale. Object recognition is then accomplished by transformation of an object from a primal sketch parameter space to the feature space of seven parameters mentioned above. This mapping can be done by means of Hough Transform.

The internal representation of a 3-D object is most conveniently represented by a depth map [27] or a needle diagram [19]. The representation consists of planar patches bounded by edge segments. Each planar patch is associated with a normal that indicates its orientation and area. For a 2-D object, the representation consists of oriented edge segments as shown in Fig. 8. The analogous 3-D object representation is also shown.

In representing shapes, one must keep in mind three different coordinate systems:

1) a three-dimensional coordinate system (x_b, y_b, z_b) which is an object-centered system.

2) a three-dimensional viewer centered coordinate system (x_v, y_v, z_v) which describes the scene being imaged.

3) an image coordinate system (x_i, y_i) which describes how the world is projected into a retina.

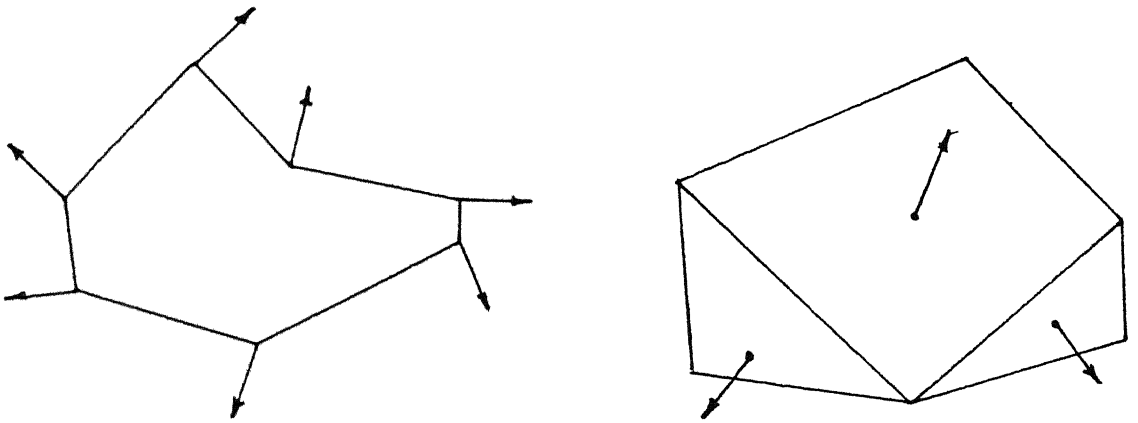


Fig. 8 :

Internal Representation

We describe below an algorithm for the detection of shape. One important assumption that has been made is that image intensities have been processed into a description that contains primitives that are like those of the internal representation.

We first study the problem of a 2-D shape recognition. The feature space comprising of parameters for scale, orientation and translation can be decoupled so as to reduce dimensionality we will discuss this aspect subsequently.

The internal representation describes the shape as a set of pairs of vectors (n_b, v_b) . The first element of each pair, n_b, is a directed edge perpedicular to the boundary of the shape. Each v_b denotes avector from the tail of n_b to the origin of an object-centered coordinate frame. Each n_b is represented as a length and an angular orientation (l_b, θ_b). v_b is represented as an offset (Dx,Dy). Ordered pairs (n_b, v_b) are elements of a set of translation offsets indexed by n_b. The value of the indexing is that for any orientation n_b one can directly find the offset v_b which specifies the position of the body-centered origin.

The algorithm consists of the following modules:

A. Detection of Scale Charge:

This consists of the following steps:

1) Form an accumulator array of possible scale factors, A $(S_{min} .. S_{max})$.

2) For each viewer normal given by (l_v, θ_v) and for each shape normal (l_b, θ_b) compute the scale s as $s := l_v/l_b$ then increment the approriate accumulator address $A(s) = A(s) + 1$.

3) Maxima in the array A correspond to instances of the shape's scale.

B. Detection of Orientation:

This consists of the following steps:

1) Form an accumulator array of possible orientation $A(0.2\pi)$.

2) For each viewer normal n_v oriented at angle θ_v, assume it is part of the shape i' it matches a body-centered normal n_b. A normal n_v is assumed to match a normal n_b if $n_v/n_b = s$, where s is the scale factor computed previously. Compute the shape's possible orientation as

$$\beta = \theta_v - \theta_b$$

and increment the appropriate accumulator array address $A(\beta) = A(\beta)+1$.

3) Maxima in A correspond to possible orientations for the shape's body-centered frame.

C. Detection of Translation:

This consists of the following steps:

1) Form an accumulator array of possible reference points $A(x_{cmin} : x_{cmax} : y_{cmin} : y_{cmax})$ initialized to zero.

2) For each edge do the following:

(a) compute θ_v, l_v

(b) calculate the possible origins, i.e., for each table entry such that

$\beta = \theta_v - \theta_b$ where the value of β is derived from module B and $l_v/l_b = s$ where the value of the scale factor, s is derived from module B.

3) Increment the accumulator array

$$A(x_c, y_c) := A(x_c, y_c) + 1$$

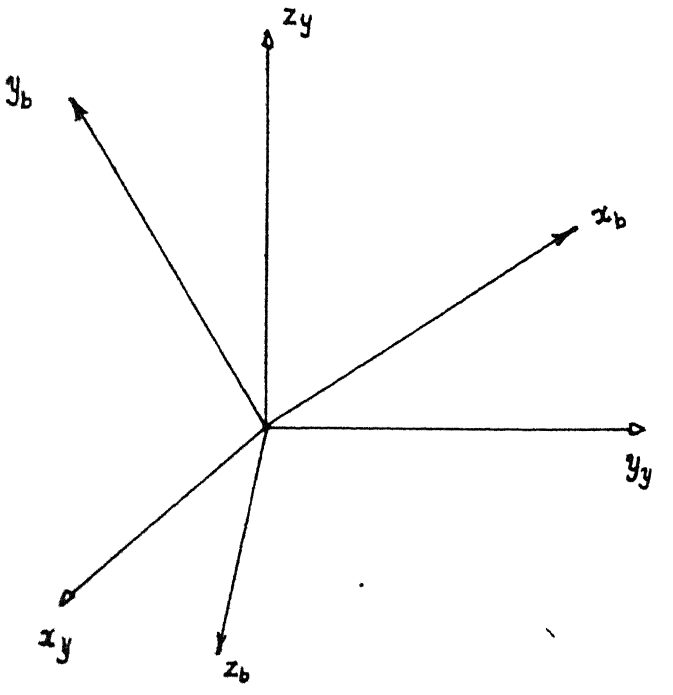
4) Possible locations for the shape's body-centered origin are given by maxima in the array A.

Detecting 3-D orientation is more difficult than 2-D orientation. This is necause, given a single surface normal, one can only ascertain the orientation of the object-centered frame with respect to the normal. The frame may still be arbitrarily rotated with respect to the normal (Refer Fig. 9). However, given a set of planes at different orientations, one can determine the final orientation parameters as the intersection of the loci of the individual normals.

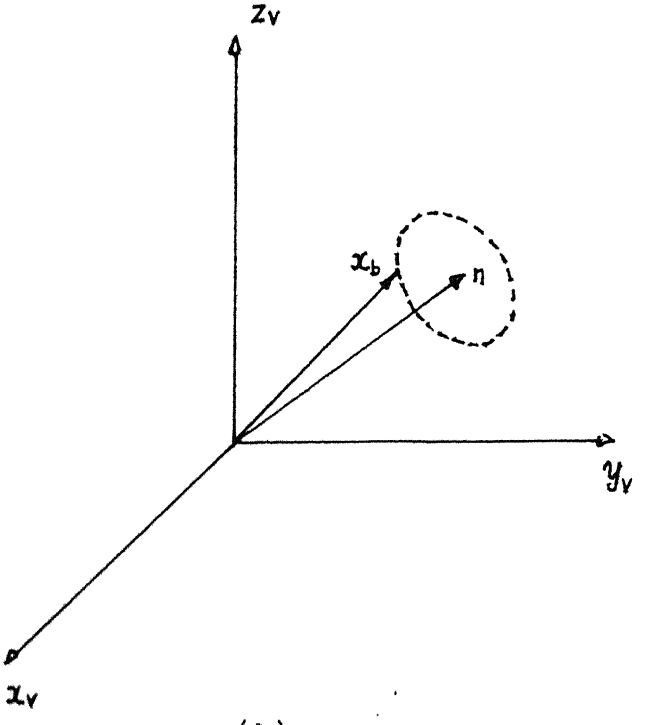
5.5 CONNECTIONIST THEORY:

The parameter net formalism aims to describe the nucleus of a connectionist theory of low-level and intermediate level vision.

Evidence from psychology suggests that the minimum time for human beings to respond to a visual stimulus is approximately 100 ms. If the time constant for a neural net



(a)
Body-Centered and viewer-centred frames treated as having same origin when considering rotation.



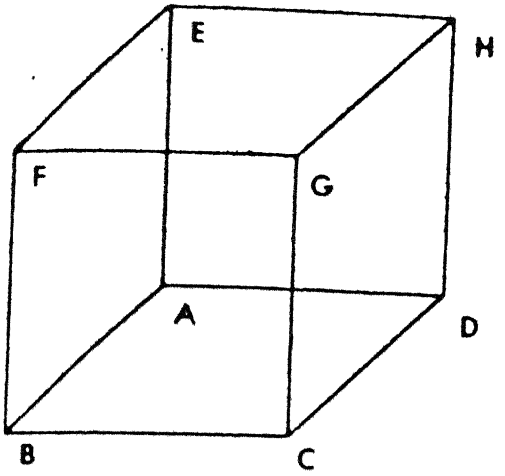
(b)
Locus of x_b defined by a match

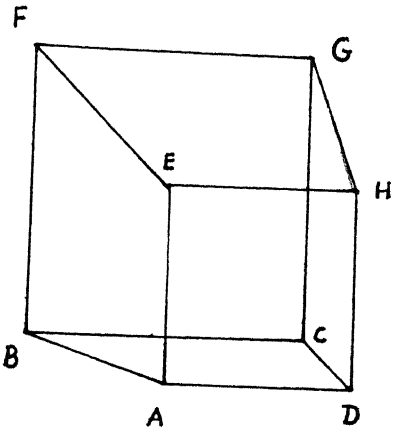
Fig. 9.

is approximately 2ms.This means that about 50 cycles are available for the complete perceptual processing and motor response in this case. The human brain has about 10^{11} neurons, any of which may be involved in the processing. If connentions must go through intermediate units, the best strategies involve on the order of the logarithm of the number of neurons. Thus there is little time to do more than link up the right neurons. Arguments like these support connectionist theory of brain processing.

The connentionist view of brain and behaviours is that all important encodings in the brain are in terms of the relative strengths of synaptic connections [14]. The fundamental premise of connectionism is that individual neurons do not transmit large amounts of symbolic information. Instead they compute by being appropriately connected to large numbers of similar units.

The connectionist computation can be well illustrated by the following example. The cube shown in Fig.10 is a famous optical illuson attributed to the Swiss naturalist L.A. Necker (1832). The cube is normally perceived with the corner a closer to the viewer, but it can also be seen as shown in Fig.10 (b) with vertex A closest to the viewer, i.e., the cube flips to another cube with the vertex A appearing closer than a. The Necker cube is interesting to psychologists because it will flip simultaneously between two views if a viewer keeps looking at it.





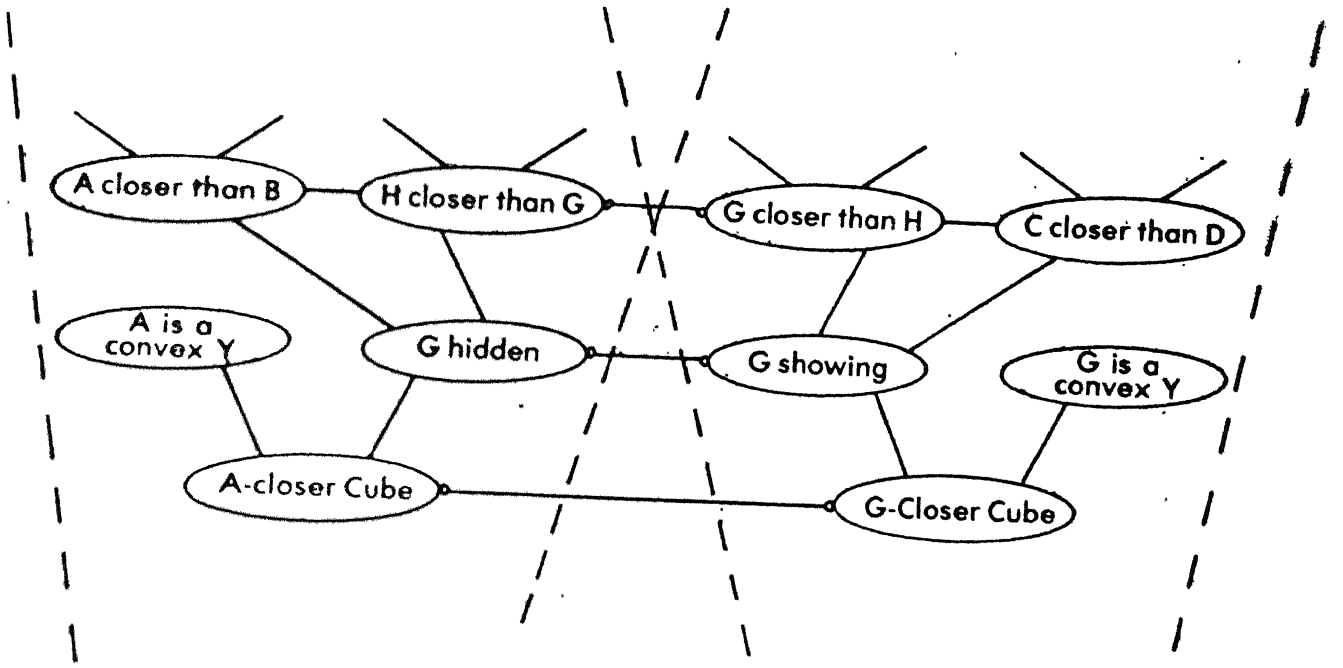


Fig. 10 The Necker cube.

The Necker cube also illustrates some of the details of the connectionist paradigm as shown in Fig. 10 (c). In the figure, each vertex is represented as a computational unit. Connections in this representation are of two types:

a) Conjuctive Connections are the connections between mutually consistent units and serve to increase each other's activity (confidence). In the figure, such connections are represented by straight lines.

b) Modifiers are connections between units that are mutually exclusive. They serve to inhibit each other's activity (confidence). Such connections are represented by lines with bubble ends.

The Necker cube phenomenon is easily explained by the connectionist representation. A sequential program running on such a slow device could probably not perform this task.

5.6 REDUCING THE SPACE REQUIREMENT:

It is currently estimated that there are about 10^{11} neurons and 10^{15} connections in the human brain and that each neuron receives input from about 10^3-10^4 other neutons. These numbers are quite large but not so large as to present any problems for connectionist theory. Since vision has a million parallel units, any algorithm requiring n^2 units would not fit. For example suppose some model called for a separate, dedicated path between all possible pairs of units in two layers in size N. It is easy to show that this requires N^2 intermediate

sites. This means, for example that there are not enough neurons to provide such a cross-bar switch for substructures of a million elements each. Two of the most important ways by which the demand on space can be reduced are as follows:

1) Functional Decomposition:

A general feature of constraint transforms is that if the computations are done completely parallel, the space required is exponential in the number of parameters. This problem can be alleviated by decomposing a high-dimensional space into lower dimensional space.

This technique was adopted in shape recognition which we have dealt with earlier. Two-dimensional shape recognition involves a four dimensional parameter network. Using 50 units for each dimension results in $(50)^4$ units. However, if the feature space is divided into three net-works-one for s, the scale factor, one for β the orientation angle and the third for the shape's body centered origin (x_b, y_b), then it results in a total of $2 \times (50) + (50)^2 = 2600$ units.

2) Coarse Coding:

The use of one unit for each discriminable feature may prove expensive in terms of demand on space. In general, for a k-dimensional feature space, the local encoding yields an accuracy proportional to the K^{th} root of the number of units.

Hinton [18] proposed dividing the feature space into large, overlapping zones and assigning a unit to each zone.

For simplicity, the zones are assumed circular, their centers are assumed to have a uniform random distribution throughout the space, and the zones used by a given encoding scheme are assumed to have the same radius.

The accuracy is proportional to the number of different encodings that are generated as we move a feature point along a straight line from one side of the space to the other. Every time the line crosses the boundary of zone, the encoding of the feature point changes because the activity of the unit corresponding to that zone changes. So the number of discriminable features along the line is just twice the number of zones that the line penetrates. The line penetrates every zone whose center lies within one radius of the line (see figure 11). This number is proportional to the radius of the zones r, and it is also proportional to their number n. Hence the accuracy a is related to the number of zones and to their radius as follows:

$$a \alpha nr$$

In general, for a K - dimensional space, the number of zones whose centers lie within one radius of a line through the space is proportional to the length of the cylinder times its (K-1) dimensional cross-sectional area which is proportional to r^{K-1}. Hence the accuracy is given by:

$$a \alpha nr^{K-1}$$

With coarse coding the accuracy is proportional to the number

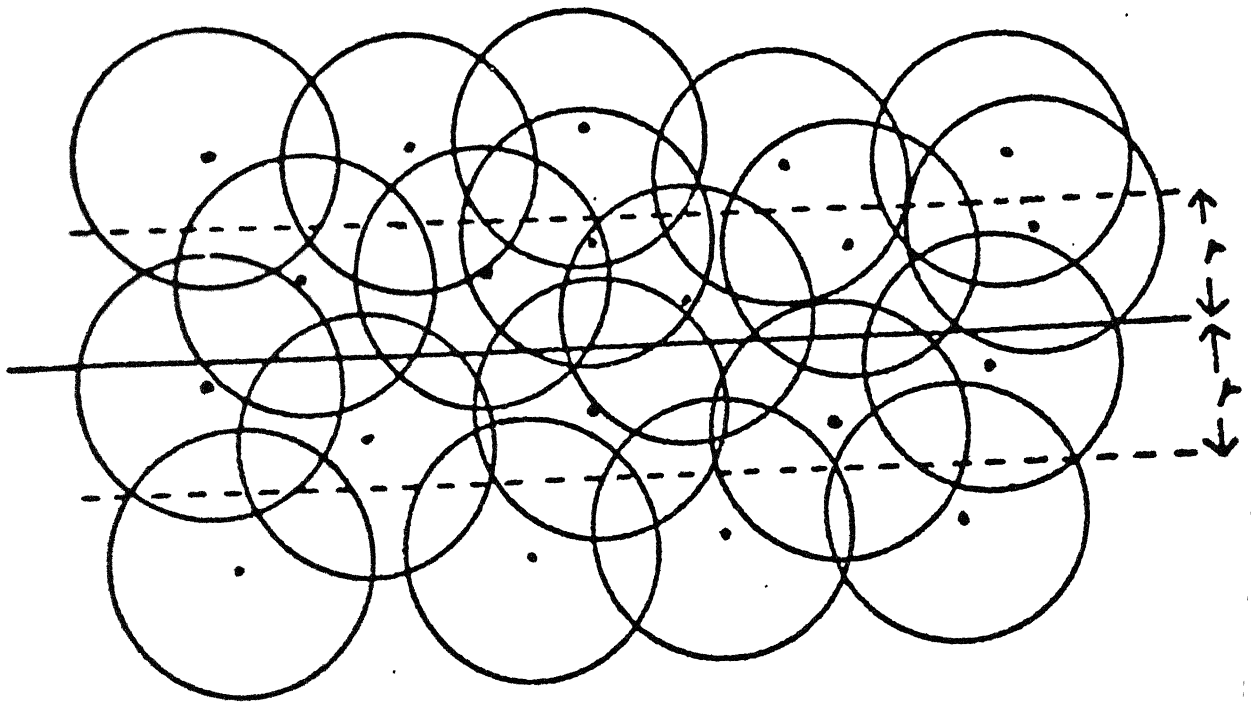


Fig. 11

of units, which is much better than being proportional to the K th root of the number.

Coarse coding is only effective when the features that must be represented are relatively sparse. If many feature points are crowded together, each receptive field will contain many features and the activity pattern in the coarse coded units will not discriminate between many alternative combinations of feature points. As a rough rule of thumb, the diameter of the receptive fields should be of the same order as the spacing between simultaneously present feature-points.

5. 7 PSYCHOLOGICAL AND BIOLOGICAL EVIDENCES:

Results of psychological test carried out by Treisman [38] can be cited in support of representation of an object (or scene) in a non-spatio-temporally indexed feature space. Treisman used displays of randomly mixed green x's and brown T's.Refer to figure 12. When subjects were asked to search for targets defined by either of two disjunctive features (say the color blue or the letters), they detected the target as quickly in a display of 30 distractors as in 1 or 5. However, searches for targets defined by conjunction of features (green and T) took time proportional to the number of letters displayed. From this we conclude that the brain processes certain primitive features in parallel but processes conjunctions of these features serially.

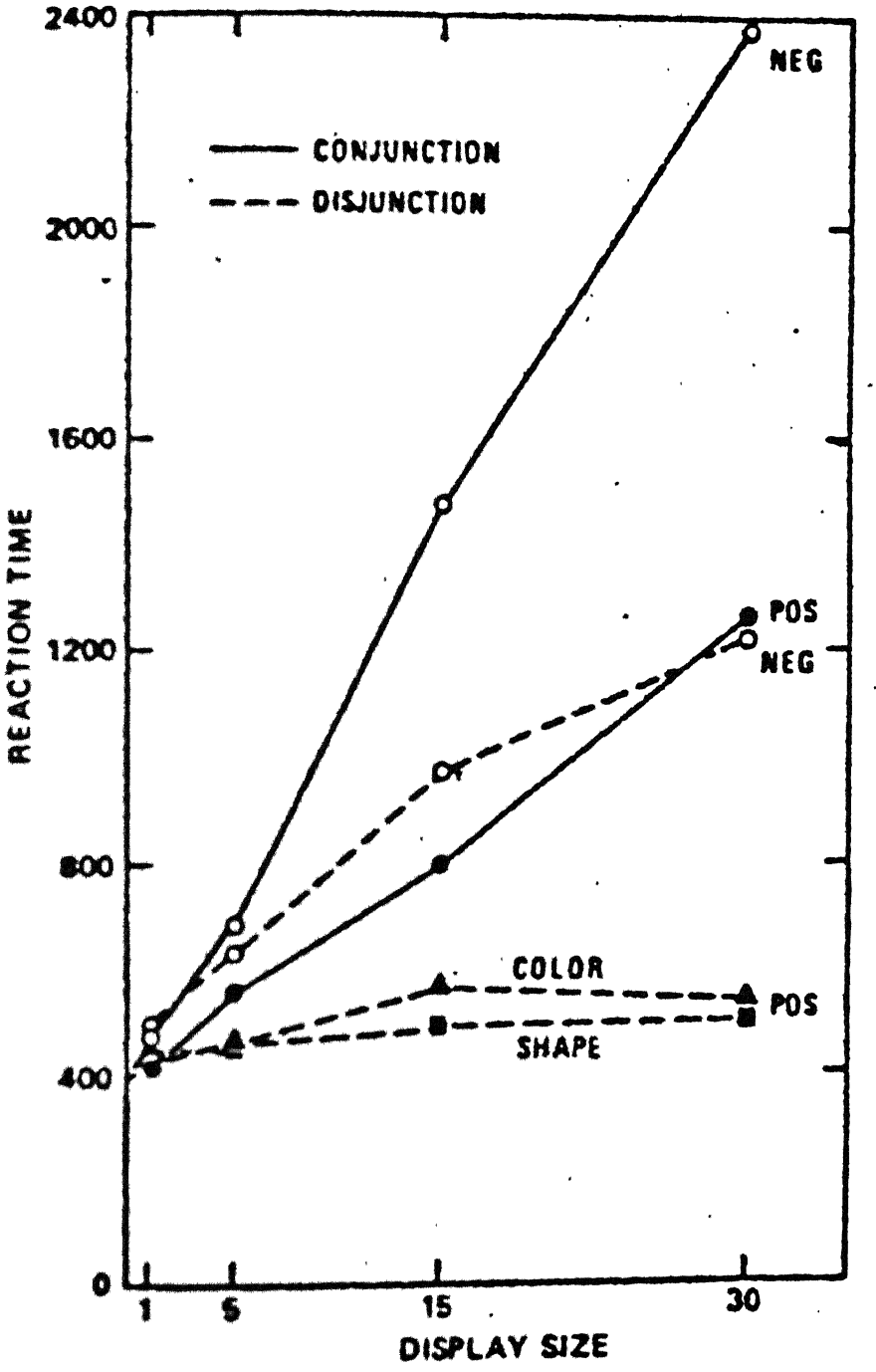


Fig. 12 Search latencies for targets defined by disjunctive features or a conjuction of features.

This probably is indicative of the fact that the representation for objects in the brain is an abstract representation that is independent of the precise point in space where the feature is located. Thus, the brain could answer questions about individual features. In contrast, questions about conjunctions of features cannot be handled by this scheme since information about spatial location is not present in the abstract representation. The brain, probably, sequentially focusses on spatial location from which the features are derived. This hypothesis is known as search light hypothesis.

Mishkin and Colleagues performed another experiment that points to this organization. They made selective lesions in different parts of a monkey's cortex. Monkeys with lesions in one area of the brain performed a feature-recognition experiment at chance levels (in a statistically random manner) but had no trouble with a spatial-location experiment. In contrast, monkeys with lesion in another area performed a spatial-location task at chance levels but had no trouble with a feature recognition experiments. This startling result shows that an important psychophysical property may have a distinct locus in animal brains.

5.8 SUMMARY:

In this chapter, we have detailed the connectionist theory of perception and the parameter net formalism. The

concept of Hough transform is introduced and recognition using this transform is described. Methods of reducing the space requirement of parameter nets are discussed . Finally psychological and biological evidences are cited as evidence in favour of such a computational structure.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION:

In the earlier chapters, we have identified the need for designing parallel computational models of vision, the pyramid and the parameter net formalism has been described in detail.

Apart from these parallel (-series) structures, there are other representations that are ' pyramid-like' and have many of the features and properties that are common to the pyramids and parameter nets. In this chapter, we will briefly discuss one such representation for shape using the Difference of Low-Pass (DOLP) transform. We, then, present a brief summary of the work carried out here. Finally we conclude by re-iterating the importance of parallel structures in modelling vision tasks.

6.2 SHAPE REPRESENTATION USING DOLP TRANSFORM:

The Difference of Low-Pass (DOLP) transform is a reversible transform which converts an image into a set of band pass images. Each band pass image is equivalent to a convolution of the image $p(x,y)$ with a band pass impulse response $b_k(x,y)$.

$$B_k(x,y) = P(x,y) * b_k(x,y).$$

provides a configuration of peaks and ridges in each band pass image which is invariant to the size of the object, except for the effects of quantization.

The description of a shape is created by detecting local positive maxima and negative minima in one dimension (ridge) and two dimensions (peaks) in each band pass image of a DOLP transform. Local peaks in the DOLP three-space define locations and sizes at which a DOLP band pass filter best fits a grey scale pattern . These patterns are encoded as symbols which serve as landmarks for matching the information in images, peaks of the same sign which are in adjacent positions in adjacent band pass images are linked to form a tree. During the linking process, the largest peak along each branch is detected. The largest peak serves as a landmark which marks the position and size of a gray-scale form. The paths of the other peaks which are attached to such landmarks provide further description of the form, as well as continuty with structures at various other resolutions.

One remarkable aspect about this representation is that the network of symbols that describe a shape are invariant to size, orientation and position of a shape.

6.3 SUMMARY OF THE WORK DONE:

We have made an extensive study of some of the parallel computational models of vision. We have examined

the need for such models and their usefulness. We have, implemented the following image operations in order to get a better appreciation of these models.

1) Segmentation of an image using an over lapping window-type pyramid. Details of this implementation has already been covered in chapter 4.

2) Object recognition using the connectionist interpretation of Hugh transform. The algorithm used for carrying out this has been detailed in chapter 5.

Given an image, its edges are detected by means of the Marr-Hildreth edge operator. For this purpose, we implemented the Marr-Hildreths edge dectection algorithm. The edge figure is then converted to an internal representation form as mentioned in chapter 5. We have, implemented the recognition of two-dimensional objects only. This method can be extended to recognition of three-dimensional objects provided the depth map (i.e., information about the depth of the surface) of the object to be recognized are known.

6.4 CONCLUSION:

All through the report, we have given arguments in favour of designing parallel models of vision. We would not like to repeat all that has been stated earlier. However, the following closing remarks will be in order.

The pyramid structure form an elegant and convenient computational structure for modelling vision tasks in parallel. The pyramids will, certainly, continue to be widely used as a computational tool for performing various image operations. But in the present situation where the trend is not merely towards designing efficient vision systems but also towards designing vision models that to some extent mimic the visual perception of living animals,the parameter nets and the concomitant connectionist theory have generated a lot of excitement andinterest. However, it is too early to pass a judgement on the impact that the parameternets will have in the efforts to design an efficient, general purpose vision system. Many issues such as stability, convergence, representation of complex concepts etc. are still unresolved in the parameter net formalism.

The question of the usefulness of designing parallel architecture models in understnading (or solving) problems in vision has raised the hackles of many vision researchers. Marr classified the issues in vision into two levels: the competence level and the performance level. The competence level specifies what is being computed and why and the performance level specifies the particular algorithms to carry out the computation. Marr argued that the theory of computation must prceede the design of algorithms and that vision researchers must not confuse the two. Many vision researchers still believe that all efforts should be concentrated on tackling issues

at the competence level only and are highly sceptical of the usefulness of designing parallel computational models. But recent research works have shown that the models themselves give rise to illuminating constraints which are useful in understanding vision tasks. There is every reason to believe that the competence issues cannot be viewed in isolation and that efforts in designing of efficient, parallel computational models should go hand in hand with efforts involved in tackling competence level issues.

BIBLIOGRAPHY

B1) Ballard, D.H., and Brown, C.M. Computer Vision, Prentice Hall, New Jersey (1982).

B2) Braddick, O.J., and Sleigh, A.C., (Eds.), Physical and Biological Processing of Images, Springer-Verlag (1983).

B3) Grimson, W.E.L., From Images to Surfaces: A Computational Study of the Human Early Visual System, MIT Press, Cambridge (1981).

B4) Hanson, A.R., and Riseman, E.M. (Eds), Computer Vision Systems, Academic Press, New York (1978).

B5) Preston, Jr., and Uhr L., (Eds.) Multicomputers and Image Processing: Algorithms and Programs, Academic Press, New York (1982).

B6) Rosenfeld., A., (Ed.), Multiresolution Image Processing and Analysis, Springer Verlag (1984).

B7) Rosenfield A., and Kak A.C., Digital Picture Processing, Academic Press.

B8) Tanimoto S., and Klinger A., (Eds.), Structured Computer Vision: Machine Perception through Hierarchial Computation Structures, Academic Press, New York (1980).

B9) Artificial Intelligence Vol. 17, (Special Issue on Computer Vision, (1981).

REFERENCES

1. Ballard, D.H., (1981), Generalizing the Hough Transform to Detect Arbitrary Shapes, Pattern Recognition, Vol.13, No.2, 111-122.
2. Ballard, D.H. (1984), Parameter Nets, Artificial Intelligence, Vol. 22, 235-267.
3. Ballard, D.H, and Sabbah, D., (1983), Viewer Independent Shape Recognition, IEEE Transactions on PAMI, Vol.5, No. 6, 653-660.
4. Ballard, D.H., and Kimball, O.A, (1983), Rigid Body Motion from Depth and Optical Flow, Computer Vision Graphics and Image Processing.
5. Barrow, H.G., and Tenenbaum, J.M., Recovering Interinsic Scene Characteristics from Images in [B4].
6. Besl, P.J. and Jain, R.C. (1985), Three Dimensional Object Recognition, ACM Computing Surveys, Vol. 17, No.1.
7. P.J. Burt, T.H. Hong and A.Rosenfeld (1981). Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchial Computation, IEEE Transactions on SMC, 802-809.
8. Clowes, M.B (1971), On Seeing Things, Artificial Intelligence, Vol. 12, 79-116.
9. Cohen, P.R., and Feigenbaum, E.A., The Hand Book of Artificial Intelligence (Vol.3), Pitman Books Limited, London (1984), 125-321.

10. Crowley J.L., A Multiresolution Representation for Shapes in [B6], 169-189.

11. Duda, R.O., and Hart P.E. (1972) Use of the Hougn Transform to Detect Lines and Curves in the Picture, Comm. ACM Vol. 15, No.1, 11-15.

12. Falk, G., (1972), Interpretation of Imperfect Line Data as a Three Dimensional Scene, Artificial Intelligence, Vol.3, 101-144.

13. Feldman, J.A. and Yakimovsky, Y. (1974), Decision Theory and Artificial Intelligence: A Semantics-based Region Analyzer, Artificial Intelligence, Vol. 5,349-371.

14. Feldman, J.A., and Ballard, D-H., (1982), Connectionist Models and Their Properties, Cognitive Science, Vol.6, 205-254.

15. Guzman A., Computer Recognition of Three Dimensional Objects in a Visual Scene, in Aggarwal, J.K. Rosenfeld, A., and Duda, R.O., (Eds.), Computer Methods in Image Analysis, IEEE Press (1977).

16. Haralick, R.M. and Shapiro, L.G. (1985), Image Segmentation Techniques, Computer Vision Graphics and Image Processing, Vol. 29, No.1.

17. Henderson, T.C., (1983), Efficient Three Dimensional Object Representation for Industrial Vision Systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, 609-617.

18. Hinton, G.E., (1984), Distributed Representations, Technical Report, CMU-CS-84-157.

19. Horn, B.K.P. and Ikeuchi, K., Numerical Shape from Shading and Occluding Boundaries in [B9].

20. Huffman, D.A. Impossible Objects as Non-sense sentences in, Aggarwal, J.K. Rosenfeld, A. and Duda, Ro, Computer Methods in Image Analysis, IEE Press (1977).

21. Kanade, T., Recovery of Three Dimensional Shape of an Object from A Single View in [B9].

22. Logan Jr., B.F., (1977), Information in the zero crossing of Band pass Signals, Bell Systems Technical Journal, Vol. 56, No.4, 487-510.

23. Mackworth, A.K. (1973), Interpreting Pictures of Polyhedral Scenes, Artificial Intelligence, Vol.4, 121-137.

24. Marr, D., and Hildreth, E. (1980), Theory of Edge Detection, Proc. R. Soc. Lond. B 207, 187-217.

25. Marr, D., and Nishihara H.K. (1977), Representation and Recognition of the Spatial Organization of Three dimensional Shapes, Proc. R. Soc. Lond., B 200, 269-294.

26. Marr, D. and Poggio, T.,(1979), Theory of Human Stereo Vision, Proc. R.Soc. Lond., B 204, 301-328.

27. Marr. D., (1976), Early Processing of Visual Information Phil. Trans. Roy. Soc. B 275,483-524.

28. Pietkanin, M. and Rosenfeld., A., (1982), Gray Level Pyramid Linking as an Aid in Texture Analysis, IEEE Transactions on SMC, Vol. 12, No.3, 422-429.

29. Riseman, E.M., and Hanson, A.R., Processing Cones: A Computational Structure for Image Analysis in [B8].

30. Roberts. L., Machine Perception of Three dimensional Solids, in, Computer Methods in Image Analysis, Aggarwal, J.K., Rosenfeld, A., and Duda, R.O. IEEE Press (1977).

31. Tenenbaum, J.M., and Barrow. H.G., Experiments in Interpretation-guided Segmentation (1977), Artificial Intelligence, Vol.3, 241-274.

32. Terzopolous, D., Multiple Reconstruction of Visible Surfaces, Variational Principles and Finite-Element Representations, in [B6].

33. Treisman, A., The Role of Attention in Object Perception, in [B2].

34. Uhr.L., Psychological Motivation and Underlying Concepts, in [B8].

35. Uhr. L. (1972), Layered 'Recognition cone' Netwowks that Preprocess, Classify and Describe, IEEE Transactions on computers, Vol. 21, 758-768.

36. Ullman, S. (1979), The Interpretation of Visual Motion, MIT Press, Cambridge.

37. Van Gool et al (1985), Texture Analysis Anno , (1983), Computer Vision, Graphics and Image Processing, Vol.2", No.3.

38. Waltz, D., Generating Semantic Descriptors from Drawings of Scenes with Shadows, in, The Psychology of Computer vision. Winston, P.H., (Ed.), 19-92.